Applied Mathematical Modelling 91 (2021) 1023–1036

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

The integrated lot sizing and cutting stock problem in an automotive spring factory

Pedro Rochavetz de Lara Andrade[a,b,*], Silvio Alexandre de Araujo[c], Adriana Cristina Cherri[b], Felipe Kesrouani Lemos[b]

[a] Paraná Federal University of Technology (UTFPR), Avenida dos Pioneiros, 3131, Jardim Morumbi, Londrina-PR, Brasil, CEP 86036-370 Brazil

[b] São Paulo State University (UNESP), Avenida Eng. Luís Edmundo Carrijo Coube, 14-01, Vargem Limpa, Bauru-SP, Brasil, CEP 17033-360 Brazil

[c] São Paulo State University (UNESP), Rua Cristóvão Colombo, 2265, Jardim Nazareth, São José do Rio Preto-SP, Brasil, CEP 15054-000 Brazil

ARTICLE INFO

Article history:
Received 26 May 2020
Revised 13 October 2020
Accepted 18 October 2020
Available online 31 October 2020

Keywords:
One-dimensional cutting stock problem
Lot sizing problem
Mathematical modeling
Column generation
Automotive spring factory

ABSTRACT

In this paper, a manufacturer of automotive springs is studied in order to reduce inventory costs and losses in the steel bar cutting process. For that, a mathematical model is proposed considering parallel machines and operational constraints, besides the demand, inventory costs and limits for items and final products. This one-dimensional integrated lot sizing and cutting stock problem is solved by a price-and-branch approach, using column generation and a method for obtaining integer solutions. Results using real data show that the proposed method achieved, in reasonable computational time, considerably better solutions than current practice at the company. Furthermore, focused on managerial insights, tests with random data were performed to analyze the influence that different parameters have in the solution, as well as the overall performance of the method.

1. Introduction

In this paper, the production of truck springs in an automotive spring factory is addressed. The goal is to reduce storage costs together with the losses in the cutting of steel bars. The approach considers the one-dimensional cutting of objects (steel bars) into smaller items (springs), aiming at meeting a demand for items and final products (spring bundles), which result from the assembly of the items. The studied factory is Fama Springs (*Molas Fama*), founded in 1960 and based in Apucarana, PR, Brazil. The company focuses production on the automotive industry, manufacturing components for the suspension of small, medium and large vehicles.

In general, coil springs are used for car suspension. However, the springs used in the suspension of trucks are different. They are composed of overlapping cut steel items to provide cushioning for these truck systems. So, steel bars are cut lengthwise to produce the various items (springs) that make up each type of truck spring bundle.

According to Torres [1], the Brazilian automobile industry emerged around 1900, in the beginning; the focus was on the production of trucks and utility vehicles. The production of passenger cars only reached significant values in the national

* Corresponding author.
E-mail addresses: pedro.rochavetz@gmail.com (P.R.d.L. Andrade), silvio.araujo@unesp.br (S.A. de Araujo), adriana.cherri@unesp.br (A.C. Cherri), felipeklemos@gmail.com (F.K. Lemos).

https://doi.org/10.1016/j.apm.2020.10.033
0307-904X/

scenario in the late 1960s [1]. Currently, it is the fourth largest industry in Brazil, representing more than 7% of national GDP [2]. Car production by large multinational companies strengthens the local supply of auto parts and other products, as is the case of Fama Springs, generating indirect jobs and decreasing production costs.

For companies whose production process involves the cutting of larger objects into needed items, minimizing the loss of raw material becomes important, and the cutting stock problem arises [3,4]. This is a classic problem of combinatorial optimization found in a wide range of industries, such as paper, steel, plastic, wood, springs, among others. Problems of this kind require a little variety of small items to be fully allocated to a selection of objects of fixed size, which may be identical or heterogeneous [5]. Hifi [6] states that a large number of possible cutting patterns in this problem means it is highly complex, making it difficult to reach a good quality solution.

This study also considers the lot sizing problem, in which the minimum cost of production is sought through a balance between setup and inventory costs, while meeting the demand for all items [7]. Fiorotto et al. [8] indicate the increasing interest of addressing real cases, including specific constraints of the production environment. These real world constraints increase the complexity of the problem, so that heuristics and meta-heuristics approaches can be developed to obtain good solutions in low computational time [9–11]. According to Mohammadi et al. [12], considering also the purchase of raw material in an integrated way can improve the global result.

Aiming to improve the company's practice, an integrated approach between the cutting stock problem and the lot sizing problem is proposed. Both problems are quite dependent, so the integrated approach can improve the global result, compared to treating each problem separately [13–16]. It is important to state that no optimization approach in the spring industry, similar to this one, has been found in the literature.

The main contributions of this paper lie in the mathematical model proposed to represent the practical problem in an integrated approach. The model was developed to meet the needs of the factory, considering short-term issues. Multiple time periods, parallel machines, stock limits for items and final products are considered in this model, besides machines capacities, and operational constraints such as the limit of item types by cutting patterns, thickness, and length limits in each machine. Besides the cutting process of items (springs), the assembly of final products (bundles) is also considered. According to the future research agenda proposed in the recent review of Melega et al. [16], this paper helps fill a gap in the literature, by considering multiple parallel and capacitated cutting machines, applied to an industrial sector that has not yet been explored in this context.

Regarding the solution method, the simplex method with column generation was used to solve the linear relaxation, according to Gilmore and Gomory [4], followed by a technique to obtain an integer solution. The formulation of the subproblems is new since it considers the specificities of the company; in this case, the limit of items by cutting pattern, as well as machine limitations. Note that this is a straightforward solution method with innovative aspects, which is justified because the main focus of this paper is the application and not the methodology.

The implementation, with extensive computational tests, of such a new approach applied to a factory in the Brazilian automobile sector, among the most important in the country, contributes to both current practice and to the literature. The approach is validated by solving instances with both real and random data. The solution of the spring company instance, obtaining a very significant reduction in losses (almost 50%), demonstrates the quality of the model and the relevance of the study. Solving the random instances allows for analysis of the influence of different parameters, giving a better understanding of the problem and enabling managerial insights that can further improve practical results.

This paper is divided into seven sections, these being: this introduction (1); the literature review (2); the description of the production process (3); the mathematical model proposed (4); the explanation of the solution method (5); the presentation of the results (6); and the conclusions (7).

2. Literature review

In recent years, many companies have turned their efforts to integrated approaches, looking for a global optimal solution, naturally better than optimizing isolated problems [17]. Consequently, studies on the Integrated Lot Sizing and Cutting Stock Problem (ILSCSP) have been increasing [16]. Among the reasons for this, is the great potential for economic gains in several industries, in addition to the recent advances in computation, which allows the approach to address more complex problems.

The ILSCSP basically captures the trade-off between material losses in the cutting process and inventory costs [18]. Depending on the approach, inventory costs for items, objects and/or final products are considered. Setup costs are often considereds [19]. In addition, each practical application results in specific constraints to be added to the model, such as object production, inventory limits, machine capacity, among others.

In a recent literature review of papers that deal with the ILSCSP, Melega et al. [16] found 34 studies conducted over 32 years, nine of which had been published in the last two years of the research (2016 and 2017). Using the same criteria, searching for papers from 2018 onwards, were found another six papers. Most papers reviewed dealt with applications in the paper industry [20–22] and the furniture industry [14,15,17,23].

In [16], the papers that deal with the ILSCSP were classified. The classification considers the level of integration between both problems, in two integration factors: time and production. Considering a multi-period planning horizon, inventories give the connection between the periods, which is the first integration factor. The second factor of integration regards three production levels: the purchase/production of objects; the cutting of objects; and the assembly of final products. Then, to classify each study, it is verified which of the three production levels are considered in the model and if it contemplates multiple time periods or not.

The present study considers the objects cutting process, together with the assembly of final products, over multiple periods of time. Therefore, according to the nomenclature proposed in [16], it is an ILSCSP and is classified as (-/L2/L3/M). In their review, the authors found 9 papers with this same classification. Considering the years of 2018, 2019 and 2020, only the paper of Melega et al. [24] was found with this specific classification.

Considering the papers that approach the same integrated problem (-/L2/L3/M), Ghidini et al. [25] proposed a mathematical model for the ILSCSP that arises in a small furniture factory. In order to solving using the Simplex method with column generation, the model was simplified, disregarding the cutting machine setup costs. Two instances with real data were solved, and the authors highlighted the importance of considering practical aspects, such as capacity limits and demand fluctuation for the solution's applicability.

A heuristic method, based on Lagrangean relaxation, to solve an ILSCSP in a furniture industry is presented in [14]. The model considers guillotine cutting in two stages, in addition to setup costs for both problems. Shortly after that, in [17], the model was adjusted, considering item inventory costs and disregarding final product setup. These changes allowed the solution of the model through the Simplex method with column generation.

In [23], the mathematical model of the ILSCSP in a furniture industry is presented. An operational constraint specific to the furniture industry was considered, aiming to reduce the number of saw cycles. For the solution of the model, cutting patterns generated *a priori* were used and the authors suggest that a column generation approach could improve the results.

Suliman [26] formulated the ILSCSP as a non-linear integer model and proposed an algorithm to solve it. The algorithm works backwards, solving the last period first. It is not a complete integration, as it solves the lot sizing first, and then the cutting stock problem. The authors tested the method with a practical example in the aluminum industry, in addition to fictitious instances.

In two sequential papers, Alem and Morabito [27,28], studied a small furniture factory and strategies to deal with the risks arising from uncertainty scenarios. In [27], robust models are proposed for the ILSCSP that consider uncertainties in several parameters, such as production and storage costs, and demand. The model was tested with real data and simulated instances. In [28], a deterministic model is presented, which considers, among other constraints, setup costs in the cutting and drilling sector. The authors analyze different models of two-stage stochastic programming to make decisions on risk-neutral or risk-averse strategies.

Vanzela et al. [15] proposed a mathematical model to contemplate this integrated approach in a small furniture factory. In order to obtain a practical feasible solution, saw cycles are considered in capacity constraints. The authors compare the integrated model solution with the simulation of factory practice, where decisions are made separately.

The work in [19] is essentially theoretical, focusing on the performance of techniques for solving the ILSCSP. The authors propose a new progressive selection algorithm, in addition to two Dantzig-Wolfe decomposition approaches with column generation. The authors claim that the proposed methods are computationally feasible and capable of improving the results in comparison with other methods.

In [24], the authors consider the ILSCSP with sequence-dependent setup times and setup costs, i.e., the cutting patterns must be sequenced in order to obtain a solution for the problem. The solution method uses column generation and the integer problem is solved by decomposition approaches. Computational results, with randomly generated instances, are presented.

Finally, considering papers that approach this integrated problem in a single period (-/L2/L3/S), Farley [29] studied a textile factory, creating two models to describe it, a quadratic and an integer. Both models consider several specificities at the company being studied, such as setup costs in the cutting machines, and minimum and maximum production levels of final products. Lemos et al. [30] studied the integration of a one-dimensional cutting stock problem in an environment of multiple manufacturing modes, an unexplored approach. The main goal is to adapt the model to solve the practical problem of a concrete pole factory. In addition to solving the real problem, the proposed model is tested with several instances of random data. Table 1 shows the summary of information about each article:

As can be seen from Table 1, most papers deal with the furniture industry in a two-dimensional approach [14,15,17,23,25,27,28]. All the studies, except [19] and [24], similar to the present study, have as main objective the solution of a practical application, considering several real word constraints, and also carrying out computational tests with fictitious instances. Futhermore, the Simplex method with column generation is a technique widely used in these approaches [15,17,19,24,25,30].

We could not find any mathematical model that considers the same aspects that we are considering in this paper. The most similar is Gramani et al. [17], but the authors only consider the linear relaxation of the model. Moreover, they do not consider parallel machines and their operational constraints, which make our subproblems to generate columns totally different from the subproblems considered in [17].

3. Description of the production process

The problem studied in this paper considers an integrated approach between the lot sizing problem and the one-dimensional cutting stock problem found in the truck unit at Fama Springs. The goal is to reduce storage costs together with the losses in the cutting of steel bars.

Table 1
Classification of the related papers from the literature.

Authors	Application	Dimensionality	Periods
Farley (1988) [29]	Textile	Two-dimensional	Single
Ghidini et al. (2007) [25]	Furniture	Two-dimensional	Multiple
Gramani et al. (2009) [14]	Furniture	Two-dimensional	Multiple
Gramani et al. (2011) [17]	Furniture	Two-dimensional	Multiple
Santos et al. (2011) [23]	Furniture	Two-dimensional	Multiple
Suliman (2012) [26]	Aliminium	One-dimensional	Multiple
Alem and Morabito (2012) [27]	Furniture	Two-dimensional	Multiple
Alem and Morabito (2013) [28]	Furniture	Two-dimensional	Multiple
Vanzela et al. (2017) [15]	Furniture	Two-dimensional	Multiple
Wu et al. (2017) [19]	General	One-dimensional	Multiple
Melega et al. (2020) [24]	General	One-dimensional	Multiple
Lemos et al. (2020) [30]	Construction	One-dimensional	Single

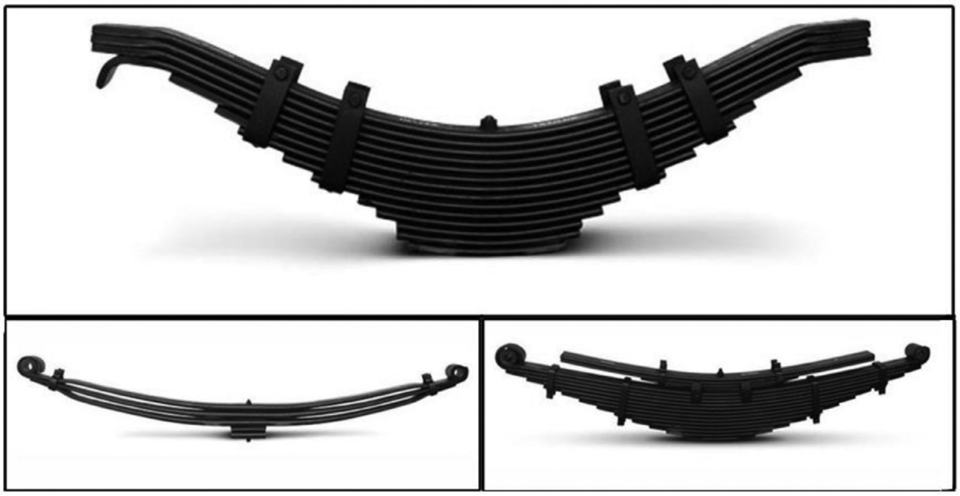

Fig. 1. Example of bundles of truck springs.

The cutting process consists of cutting objects (steel bars) into specific sized items (springs) which are used in the manufacture of final products (bundles). The bundles are composed of a set of overlapping springs, as shown in Fig. 1.

The company has three machines to carry out this process: an eccentric press that can produce 1560 items per day and cuts thicknesses up to 20 mm; one metal cut-off grinder disk with a production capacity of 1560 items per day, which cuts up to 35 mm thickness; and an automatic cutting assembly with a production capacity of 1056 items per day, cutting items up to 30 mm thick.

Another operational constraint of these machines is the limit of different types of items that can be used in a given cutting pattern (how the object is cut into smaller items). The automatic machine can cut up to 4 different types of item per cutting pattern, while the other machines accept cutting patterns up to a limit of 3 different types of item. Also, the automatic machine has a limitation in the minimum and maximum length of items that it can produce (from 500 mm up to 2000 mm).

It is worth noticing that there are few adjustments to be made on manual machines when changing the cutting pattern, setup times are short when compared to the cutting time. It occurs in changing the types of items that make up a cutting pattern, and not necessarily in changing the type of bar to be cut. For the automatic machine there is a short electronic adjustment time for each change of bar type or cutting pattern. But this time can be carried out externally, that is, while the last units of the previous lot are being cut. Based on this discussion setups will not be considered in the mathematical model presented in the next section.

The elaboration of the cutting patterns for the steel bars aims to maximize the use of the object in producing the necessary items. Following company rules, if the cutting pattern uses 95% or more of the object, it is accepted, otherwise, another cutting pattern must be drawn. As the cutting patterns are produced manually, most of the times that the homogeneous cutting pattern (where only one type of item is produced from one object) generates a loss of less than 5%, it is used. In practice, the average loss practiced by the company ranges from 4.5% to 5%, and the leftover steel is sold off by weight with a disregardable cost.

Demand may occur for both unit springs and spring bundles. Sales history of each product, and inventory data such as: current stock; committed stock; minimum and maximum stock are used for the production decision. The production strategy is of production for stock (Make-to-Stock), and the minimum and maximum stock levels defined for items and final products with high demand are higher than those for low demand. For some specific cases with very low demand, the stock should be as small as possible. Daily, the person in charge checks the inventory data and decides which springs and bundles will be produced and in what quantities.

Fama Springs has high stock levels and low inventory costs, for objects, as well as for items. So, the reduction of inventory levels is not necessarily an advantage for this company. In this way, the main gain from this practical application, from the company's point of view, occurs by reducing material loss in the cutting process. Taking this into account, as can be seen later in Section 6.1, the inventory costs in the real data instance are quite low, except for items with very low demand, for which the company does not want to keep any stock. Even though the inventories of items and objects are high, the company maintains low inventories of final products. The intention is to keep items in stock and use them to assemble final products only when necessary, since the items can also be sold separately.

It is important to mention that the unit of time for production is one day. At the end of each week, production is planned for the coming week. However, this planning can be modified daily to include client orders generated on the previous day. Only after this change the production of one day is finally fixed.

Regarding the purchase of steel bars, supplier delivery time is long, varying for each order and taking up to six months depending on the case. The company purchases standard sized bars (width, thickness and length). In total, 320 types of bar are bought, ranging from about 1 m to 7 m in length and also varying in terms of the percentages of carbon, molybdenum, vanadium, niobium and aluminium in the steel.

These differences alter the specifications of the springs to be produced, so each type of object can only be used to produce specific types of items that match its characteristics. Therefore, there are subgroups formed by object types and those item types that have similar characteristics. Each object can only be part of one subgroup and can produce any item that is also in that subgroup. It cannot produce any item from another subgroup.

In view of all the characteristics described, it was decided to address the problem in the short-term issues, considering the company's cutting machines and their specificities. Since object purchases have a long delivery time and, in this approach, a period corresponds to one day, object issues were not considered. This is a viable consideration since the high level of inventories of objects kept by the company, means, on a daily horizon, any production decision can be put into practice.

4. Mathematical model

The proposed mathematical model considers cutting objects (steel bars) into items (springs) and assembling items into final products (bundles). Both items and final products have their own demands, inventory costs and limits. Parallel machines are also considered, as well as, their operational constraints, which will appear in the subproblem. Each term used in the mathematical model is defined as follows:

Indexes:
- $i = 1, ..., I$: number of item types;
- $k = 1, ..., K$: number of object types;
- $j = 1, ..., N_K$: each N_k represents the number of cutting patterns of object k;
- $p = 1, ..., P$: number of final product types;
- $f = 1, ..., F$: number of cutting machines;
- $t = 1, ..., T$: number of periods in the planning horizon.

Parameters:
- dr_{it}: demand of item type i in period t;
- dp_{pt}: demand of final product type p in period t;
- α_{ijk}: quantity of item type i produced by cutting pattern j from object k;
- z_{ip}: quantity of item type i in one unit of final product type p;
- cap_{ft}: production capacity (in number of items) of the machine f in period t;
- c_{jkft}: cost of cutting (mm of bar) of an object type k according to the cutting pattern j on machine f in period t;
- cr_{it}: storage cost (in mm of bar) of item type i in period t;
- cp_{pt}: storage cost (in mm of bar) of final product type p in period t;
- $rmin_i$: minimum stock of item i;
- $rmax_i$: maximum stock of item i;
- $umin_p$: minimum stock of final product p;
- $umax_p$: maximum stock of final product p;
- L_k: length of object k;
- l_i: length of item i.

Decision Variables:
- x_{jkft}: number of objects type k cut according to cutting pattern j on machine f in period t;
- y_{pt}: number of final products type p produced in period t;
- r_{it}: number of items type i in stock at the end of period t;
- u_{pt}: number of final products type p in stock at the end of period t.

The mathematical model is as follows:

$$Min \sum_{t=1}^{T}\left(\left(\sum_{k=1}^{K}\sum_{j=1}^{N_k}\sum_{f=1}^{F}c_{jkft}x_{jkft}\right)+\sum_{i=1}^{I}cr_{it}r_{it}+\sum_{p=1}^{P}cp_{pt}u_{pt}\right) \quad (1)$$

Subject to:

$$\sum_{k=1}^{K}\sum_{j=1}^{N_k}\sum_{f=1}^{F}\alpha_{ijk}x_{jkft} + r_{i,t-1} = dr_{it} + r_{it} + \sum_{p=1}^{P} z_{ip}y_{pt},\ \forall\ i,\ t, \tag{2}$$

$$y_{pt} + u_{p,t-1} = dp_{pt} + u_{pt},\ \forall\ p,\ t, \tag{3}$$

$$\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{j=1}^{N_k}\alpha_{ijk}\ x_{jkft} \le cap_{ft},\ \forall\ f,\ t, \tag{4}$$

$$rmin_i \le r_{it} \le rmax_i,\ \forall\ i,\ t, \tag{5}$$

$$umin_p \le u_{pt} \le umax_p,\ \forall\ p,\ t, \tag{6}$$

$$x_{jkft} \in Z_+,\ y_{pt} \in Z_+,\ r_{it} \in R_+,\ u_{pt} \in R_+,\ \forall\ i,\ k,\ j,\ f,\ p,\ t. \tag{7}$$

The cost of cutting an object k with cutting pattern j, on machine f, in period t, represented by c_{jkft}, is equivalent to the loss of material, in millimeters, generated by this cutting pattern, that is:

$$c_{jkft} = L_k - \sum_{i=1}^{I}\alpha_{ijk}l_i,\ \forall\ k,\ j,\ f,\ t. \tag{8}$$

In the model (1) - (7), the objective function (1) minimizes costs with loss of material in all periods (days), as well as the costs of storing items (springs) and final products (bundles). Constraints (2) ensure that the demand for all items in all periods is satisfied. Under these restrictions, for all items in all periods, the quantity available in stock at the beginning of the period ($r_{i,t-1}$), plus the quantity produced for this item ($\sum_{k=1}^{K}\sum_{j=1}^{N_k}\sum_{f=1}^{F}\alpha_{ijk}x_{jkft}$) must be equal to the sum of the demand for this item (dr_{it}), the quantity used in the production of all the bundles types ($\sum_{p=1}^{P} z_{ip}y_{pt}$), and the quantity left in stock for the next period (r_{it}). In (3) the demand for final products is satisfied, since for all products p in all periods t, the quantity in stock at the beginning of the period ($u_{p,t-1}$) plus the quantity produced (y_{pt}), should be equal to product demand (dp_{pt}) plus quantity left in stock for next period (u_{pt}). Constraints (4) ensure the production capacity of each cutting machine is respected in all periods. Constraints (5) and (6) establish the minimum and maximum stocks for items and final products, respectively. The initial stocks r_{i0} and u_{p0} are parameters that vary with each instance and can assume any value between the established limits. Finally, in (7), the domain of the decision variables is defined.

5. Solution Method

The method used to solve the linear relaxation of the model presented in Section 4 is the Simplex with column generation [6]. A computational package was used to obtain integer solutions. Similar approaches were used by Vanzela et al. [15], Poldi and de Araujo [21], among others. In this section, the application of this method is explained in detail.

The model has a large number of integer variables, x_{jkft} and y_{pt}, making it difficult to obtain the optimal solution. And so, these variables have their integrality constraints relaxed (using x_{jkft} and $y_{pt} \in R_+$) and the Simplex method with column generation is used. This model is usually called the "master problem".

Initially, the master problem is solved considering only the homogeneous maximal cutting patterns (patterns containing only one item type in as much quantity as possible), which generates a dual variable value for each constraint. Then the values of the dual variables referring to Constraints (2) and (4) are used to compose the objective function of the subproblems, presented in (9). The subproblems are mathematical models of the knapsack problem, formulated so that new columns (cutting patterns) are generated and thus making optimization of the master problem.

At each iteration, the number of subproblems solved is equal to the product of the number of objects (K), machines (F) and periods (T). The items assigned to a cutting pattern are only those that can be produced from a given object k and by a given machine f, ensuring that the cutting patterns generated are technically feasible.

Adaptations were made to the formulation of the subproblems to consider the specificities of this particular company. As a subproblem is generated for each object, each machine and each period, resulting in a cutting pattern, the indexes k, j, f and t are fixed in each subproblem. Therefore, the terms L_k, π_{ft} and $imax_f$, given below, are treated as constants. The variable α_{ijk} has the fixed indexes j and k, with only index i varying. In addition to those presented previously, the terms used in the mathematical model of the subproblems are:

Sets:
G_k: Set of items that can be produced from object k;
H_f: Set of items that can be produced on machine f.
Parameters:
π_{it}: the dual variable of Constraint (2), referring to item i in period t;
π_{ft}: the dual variable of Constraint (4), referring to machine f in period t;
$imax_f$: maximum number of different items for a cutting pattern on machine f;
M: sufficiently large value.
Variables:
α_{ijk}: quantity of item i produced by cutting pattern j from object k;
β_i: whether or not item i is present in the cutting pattern.

The mathematical model of the subproblems (for each $k = 1, ..., K$, $f = 1, ..., F$, and $t = 1, ..., T$) is described below:

$$Min\ L_k - \sum_{i \in G_k \cap H_f} \alpha_{ijk}\left(l_i + \pi_{it} + \pi_{ft}\right), \tag{9}$$

Subject to:

$$\sum_{i \in G_k \cap H_f} \alpha_{ijk} l_i \le L_k, \tag{10}$$

$$\sum_{i \in G_k \cap H_f} \beta_i \le imax_f, \tag{11}$$

$$\alpha_{ijk} \le M\beta_i, \ \ \forall\ i \in G_k \cap H_f, \tag{12}$$

$$\alpha_{ijk} \in Z_+, \ \ \beta_i = \{0;\ 1\}, \ \ \forall\ i \in G_k \cap H_f, \tag{13}$$

In (9)-(13), the objective function (9) minimizes the relative cost of the generated column. Constraints (10) ensure that the sum of the lengths of items in the cutting pattern is not greater than the length of object k. In (11), the number of different items that can make up a cutting pattern on machine f is limited. Furthermore, constraints (12) link the decision variables, ensuring that, if the item i is not in the cutting pattern ($\beta_i = 0$), the variable α_{ijk} for this item is equal to zero. Otherwise, ($\beta_i = 1$), the variable α_{ijk} can be any value, provided that the constraints are respected. Finally, (13) defines the domain of variables.

To strengthen the linear relaxation, the value of M should be as small as possible. From (10), it is known that $\alpha_{ijk} \le L_k/l_i$. Therefore, the value for M is equal to: $L_o/l_p + 1$, L_o being the length of the largest object, and l_p the length of the smallest item.

When the relative cost of the column generated by a subproblem is negative, it means that the column improves the quality of the current solution, and it should be added to the master problem. The lower the value, the greater the improvement generated by this column. It is known adding more columns to the master problem can improve the quality of the final solution, and it also can lead to fewer iterations required to reach it. On the other hand, adding more columns increases master problem (relaxed and integer) processing time. Additionally, as it can be seen in Section 6, the most significant processing time in the largest instances, especially the real data instance, is required to solve the relaxed master problem, and the integer master problem, rather than the subproblems.

Computational time is also a scarce resource, either because of the risk of memory overflow in very long processing or, in real cases, the need for a solution in the right time for its utilization. Therefore, to reduce the computational time of the master problem without reducing the quality of the final solution, a strategy was created. It consists in selecting only better columns to add in the master problem at each iteration, those with the lowest relative costs. After solving $K * F * T$ subproblems in an iteration, the average of the negative relative costs is calculated. Then, only columns generated by subproblems with relative cost less than or equal to the value of the mean multiplied by 0.1 are added to the master problem. This value (0.1) was obtained based on initial tests. Since we are dealing with negative values, the lower the value, the more columns are added.

Fig. 2 shows a flowchart that summarizes the proposed solution method. In the first iteration, homogeneous cutting patterns are generated and the relaxed master problem is solved. The dual variables are used in the objective function of the subproblems to be solved (K * F * T subproblems at each iteration), the generated columns are added to the master problem according to the explained criteria. It is important to note that the columns considered for calculating the mean are only those that obtain a negative relative cost. Thus, the master problem is solved considering the new columns, starting a new iteration.

When, after solving all subproblems, no generated cutting pattern obtains a negative reduced cost, it means that the optimal solution of the linear relaxation of the master problem was reached. Then a procedure is applied to determine an integer solution. In this procedure, the homogeneous columns initially generated, and the columns added during the iterations are used in the master problem, which is solved considering the integrality constraints. Finally, the gap is calculated, which is the difference in percentage between the integer solution and the optimal solution of the linear relaxation.

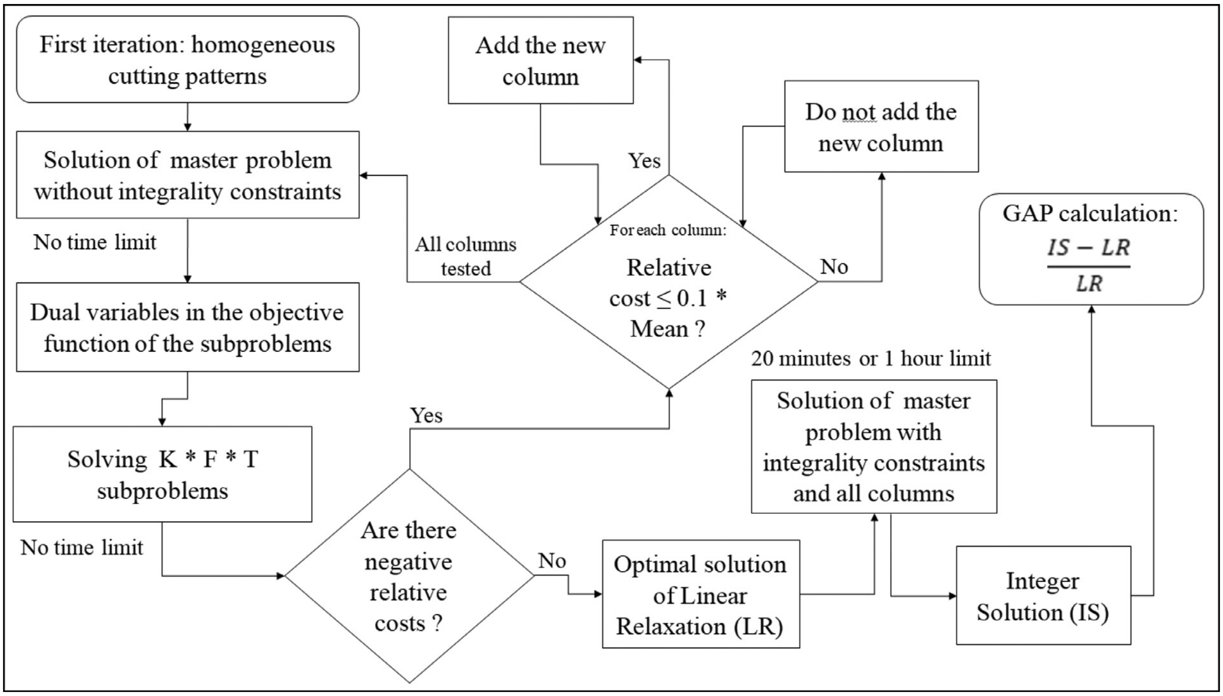

Fig. 2. Flowchart of the solution method.

It is possible to observe, in Fig. 2, that no processing time limit was defined in the column generation, but in the solution of the master problem with integrality constraints, the time limit was set at 20 minutes for the random data instances and 1 hour for the real instance.

6. Computational results and discussion

In this section, the results of applying the proposed model (1)-(7) are presented. The results include real data from the industry, in Section 6.1, and randomly generated data, in Section 6.2. The detailed data are available online on the GitHub platform, through the following link. https://github.com/prochavetz/AMMOD-D-20-01547.

Optimization Programming Language (OPL) modeling language was used to build the mathematical model, and the CPLEX solver (version 12.10) was used to solve it. Visual Basic for Applications was used to handle the company data, generate the file to be processed by the OPL, as well as to analyze the results. The computer used to solve the instances has an Intel Core i7, 64 bit processor with 16GB RAM.

6.1. Real data

The instance solved considers the weekly production of the company (five periods), for a given specific week, with 53 types of objects used for the production of 176 types of items, divided into 45 subgroups (each subgroup contains compatible items and objects). As explained, the company has three cutting machines (two manual and one automatic) and their production capacity, limitations in terms of item thickness and length, and the limited types of items per cutting pattern were taken into consideration.

Although in practice, the daily inventory costs of most of the items or final products are not relevant, they were considered in the model to balance inventory levels, preventing them from rising excessively. As explained in Section 3, according to company specifics, inventory costs are quite low for almost all items. Items with high demands had inventory costs per period defined as 0.5% of their lengths. For items with low demand, the established costs were 1% of their lengths. In addition, for items with very low demand, the inventory cost is set to 50% of their lengths. Inventory costs for final products were defined as the sum of inventory costs for all items that comprise it. These definitions were made in consultation with the production manager. Thus, the percentage values established for each item, as well as the calculation of inventory costs for final products were considered quite reasonable by the company.

The total demand for the week studied is 13,305 items and 221 final products, which need 1,779 items to be produced.The total production capacity is 20,880 items. The average number of item types per subgroup is 3.9, and the largest subgroup has 53 item types. There is only one item type in the smallest subgroup. The average number of object types per subgroup is 1.2, and the number of object types per subgroup ranges from 1 to 3. Considering stocks of all types of items,

Table 2
Spring factory data.

Parameter	Minimum	Maximum	Mean	Standard Deviation
Object Length	1,200	7,000	6,052	875.3
Item Length	300	2,220	1,178	475.6
Item Demand	4	1,200	76.0	148.6
Final Product Demand	6	82	31.6	23.4
Item by Subgroup	1	53	3.9	7.8
Object by Subgroup	1	3	1.2	0.48
Item Stock Range	31	3,030	164.0	427.9
Final Product Stock Range	30	52	38.0	6.5
Item Stock Cost	0.5%; 1%; 50%		16.9%	23.2%
Production Capacity	138% of the need for items			

Table 3
Results of the model using real data.

Factor		Value
Solution	OF (mm)	1,574,639
	Gap	9.83%
Loss	Objects Used	3,109
	Total Cut (mm)	19,767,110
	Loss (mm)	470,216
	Loss %	2.38%
Item Stock	Usage %	10.86%
	Stock Cost (mm)	602,049
	Cost %	3.18%
Bundles Stock	Usage %	13.16%
	Stock Cost (mm)	502,374
	Cost %	1.68%

the average difference between the maximum stock and the minimum stock is 164 items, with the smallest difference being 31 items and the largest, 3,030 items. Table 2 presents the relevant data in this instance with the minimum, maximum and mean values, as well as standard deviations.

Table 3 shows the quality of the solution obtained using the proposed approach. The value of the objective function (OF) and its percentage difference in relation to the optimal solution of the linear relaxation (Gap) are presented. This table also presents the number of objects used, the total length of objects cut, the total length of the losses (both in millimeters) and the percentage of losses. Finally, the inventory costs (Stock Cost), the inventory usage index (Usage %) and the inventory cost index (Cost %) are shown for both items and bundles.

To find, for example, the value of the inventory usage index of items for each instance, the inventory range is calculated ($range_i = rmax_i$ - $rmin_i$) for all item types. The average stock occupation of each item type is also calculated ($rocp_i = rmean_i$ - $rmin_i$), where ($rmean_i = \frac{\sum_{t=1}^{T} r_{it}}{T}$), for all item types. So, the inventory usage index is equal to ($\frac{\sum_{i=1}^{I} rocp_i}{\sum_{i=1}^{I} range_i}$. 100). The calculation is the same for the final products. The same criterion is used to calculate the inventory cost index, but referring not to the number of units in stock, but the cost of those units in stock.

Taking into account the size of this instance and its characteristics, the gap in this solution can be consider a good result. In Section 6.2.2, it is clear that the biggest gaps occur in larger instances with small items, exactly the characteristics of this real instance.

At the spring company, the loss from the cutting process is measured in kilograms of steel lost, which means that the loss in objects with larger width and thickness, is more significant than the loss from smaller objects. During production in the week being studied, the loss in practice at the company was 4.75%, losing 6,657 kg of steel after cutting 3,308 objects whose total weight was 140,207 kg. The proposed mathematical model use loss calculated in millimeters, but to allow the comparison, in this case, the solution was analyzed in terms of weight: 3,960 kg of steel was lost after cutting 3,109 objects weighing a total of 165,908 kg, which represents a loss of 2.39%.

Note that, besides cutting 25,701 kg more, the model lost 2,697 kg less. The proposed solution also used 199 fewer objects and, as can be seen below, produced a total of 674 more items. To compare the solution, the percentage of losses generated by the model was compared with actual company data. In view of this, it can be concluded that the loss was reduced by 49.7%, from 4.75% to 2.39%. In the actual company solution, this reduction would represent a weekly saving about 3,307 kg of steel. When analyzing the loss of the proposed solution in terms of length, as will be done with all other instances of this paper, 470,216 mm of steel were lost after cutting of objects measuring 19,767,110 mm in total, which represents a 2.38% loss.

Table 4
Computational performance of the model with real data.

Factor		Value
Number of Subgroups		45
Number of Iterations		16
Columns	Homogeneous	487
	Generated	762
Relaxed Master Time (s)	By Iteration	189
	Total	3,031
Subproblem Time (s)	By Iteration	40
	Total	637
Integer Master Time (s)		3,874
Total Time (s)		7,543

Regarding the inventory usage index, the solution reaches aceptable values both for the items (10.86%) and final products (13.16%). The values of the inventory cost index are much lower, 3.18% for items and 1.68% for final products. This large difference is made possible by the large difference in inventory costs established in the model, with inventory costs of 0.5% for some units and 50% for others. This difference between the inventory usage index and the inventory cost index also means that the model was capable of avoiding stocks of high cost, accumulating inventories mostly from low-cost units, in this case, both for items and final products.

In practice, the inventory usage index of the company was slightly higher than the proposed solution, 11.5% for the items, and 15.2% for the final products. Even if the company does not measure storage costs, and therefore does not make decisions based on them, for comparison purposes only, the actual company solution was evaluated with the same criteria, obtaining an inventory cost index of 51.2% for items and 16.8% for final products.

The solution of the proposed model uses 83.71% of the total capacity, producing 17,478 items in the five periods, meeting the demand for items and final products. By comparison, the company solution in practice used 80.48% of the capacity to produce 16,804 items. As the total need for items was 15.084 units, both solutions increase inventories. In the solution of the model, production close to the demand period allowed that, even with an increased production, the inventory cost indexes remained low. Additionally, the increase in production occurs mainly in items with low stock costs that allow good cutting patterns, consequently reducing losses. Achieving this result was facilitated by considering low inventory costs. In addition, it is important to note that even though inventories have been accumulated, there is a limitation for this increase, since all types of items and final products remain, in all periods, below the maximum stock established by the company.

Table 4 shows the computational performance of the model applied to the spring factory data. The first lines show the number of subgroups of items and objects and the number of iterations required for the solution. The number of columns, considering initial columns from homogeneous patterns (Homogeneous), and the columns generated during iterations (Generated) are given next. The total time and the time per iteration (both in seconds) spent solving the relaxed master problem and subproblems (By Iteration and Total) follow. Finally, the time needed to solve the master problem with integer constraints at the end of the run and the total computer time for all stages are shown.

As can be seen in Table 4, most of the 2 h and 6 min total computational time was used in solving the integer master problem. The processing time of the integer master problem was limited to 1 h, and the excess time (274 s) was consumed in loading the data. In the column generation, most of the processing time was spent solving the relaxed master problem. To compare the time taken to find the solution, the spring company estimates that an employee takes 42,000 s (11.7 h) per week or 8,400 s (2.3 h) per day to do this manually. A practical gain lies in the fact that the solution is automatically found, requiring only a few minutes for an employee to enter the data and export the solution.

6.2. Random data

In order to evaluate the performance of the model (1)-(7), random instances with different characteristics generated as described in Section 6.2.1 were used. These instances also allowed additional managerial insights to be obtained. The results are presented in Section 6.2.2.

6.2.1. Generation of the random data instances

These instances were divided into 18 groups, which differ in terms of the number of items and object types, inventory costs, and length of the items. The characteristics of each group are shown in Table 5:

For each of the groups, 10 instances were generated with random data, within the levels fixed above, totaling 180 instances. Each instance has, besides the number of items and objects that are in each group, four periods and three machines having the same characteristics as those used by the spring company. The other necessary information was randomly defined, based on the mean and standard deviation of the real data. The parameters considered and their ranges are shown in Table 6.

The demand for each item and final product occurs in only one of the four periods, which is also defined at random. The definition of minimum and maximum inventories for each item is based on their demand, so that limits are higher for

Table 5
Groups of instances with random data.

Random Instances		Size of the Instances					
		35 Item Types; 15 Object Types; 5 Final Product Types			70 Item Types; 30 Object Types; 10 Final Product Types		
		Item Inventory Costs					
		0-5%	10-15%	20-25%	0-5%	10-15%	20-25%
Length of Items (mm)	500-1,000	G 1	G 4	G 7	G 10	G 13	G 16
	1,000-1,500	G 2	G 5	G 8	G 11	G 14	G 17
	1,500-2,000	G 3	G 6	G 9	G 12	G 15	G 18

Table 6
Range of variation of the parameters of the random instances.

Parameter	Minimum	Maximum
Item Demand	10	300
Final Product Demand	5	40
Object Length	5,200	6,800
Item Stock Range	0; 20; 45; 80	30; 55; 90; 140
Final Product Stock Range	0 – 10	15 – 20
Item by Subgroup	1	7
Item by Final Product	4	10
Object by Subgroup	50% current subgroup, 50% next subgroup	
Production Capacity	115% of Items Demand	

Table 7
Percentage gap for the model with random instances.

Gap		Size of the Instances								**Mean**
		35 Item Types; 15 Object Types; 5 Final Products Types				70 Item Types; 30 Object Types; 10 Final Products Types				
		Item Inventory Costs								
		0-5%	10-15%	20-25%	**Mean**	0-5%	10-15%	20-25%	**Mean**	
Length of Items (mm)	500-1,000	1.03%	1.26%	1.50%	**1.26%**	2.63%	2.36%	2.05%	**2.47%**	**1.81%**
	1,000-1,500	0.50%	0.66%	0.69%	**0.62%**	0.62%	0.68%	0.83%	**0.71%**	**0.66%**
	1,500-2,000	0.45%	0.56%	0.58%	**0.53%**	0.49%	0.58%	0.67%	**0.58%**	**0.55%**
Mean		**0.66%**	**0.83%**	**0.92%**	**0.80%**	**1.25%**	**1.21%**	**1.19%**	**1.21%**	**1.01%**

stock items with high demand. The initial stock for items (r_{i0}) and final products (u_{p0}) is equal to the minimum stock $rmin_i$ and $umin_p$.

To bring the instances closer to the company's practice, the inventory levels of final products are low. The inventory cost of final products, in the same way as the real data instance, is the sum of inventory costs for all items that comprise it. Finally, it is important to state that after defining (between 4 to 10) the items that make up a final product, the quantity is also randomly defined, between 1 to 3 units of each item.

The total capacity of the machines was considered as 115% of the need for the production of items in each instance to avoid infeasibilities. The capacity is equally divided by machine and period. The criteria to define the number of objects in each subgroup is that the first object makes up the first subgroup, and then each object has 50% chance of being in the same subgroup, as the previous object, and 50% chance of forming a new subgroup.

6.2.2. Results of random instances

Table 7 below, illustrates the average gap for the 10 random instances in each of the 18 groups. It is clear that instances with the smallest items generate large gaps. This occurs because small items generate more diversities of cutting patterns. As only a part of these cutting patterns (columns) are generated, it becomes more difficult to find solutions very close to the optimal solution, compared to instances with fewer possible cutting patterns. In addition, lower gaps are achieved by smaller instances. Therefore, in a practical application with a high concentration of small items, the results from the proposed approach must be analyzed and possibly improved.

Table 8 shows the mean percentage loss for each group of instances. Since smaller items allow better combinations in the cutting patterns, instances with the smallest items generate smaller losses. Additionally, for low inventory costs, lower are the losses. With low inventory costs, stock levels may be higher, which allows better matching of items, reducing losses. Finally, note that large instances tend to have slightly large losses. Therefore, in a practical application, if possible, it is

Table 8
Percentage losses of the model with random instances.

Loss		Size of the Instances								Mean
		35 Item Types; 15 Object Types; 5 Final Products Types				70 Item Types; 30 Object Types; 10 Final Products Types				
		Item Inventory Costs								
		0-5%	10-15%	20-25%	**Mean**	0-5%	10-15%	20-25%	**Mean**	
Length of Items (mm)	500-1,000	0.76%	1.17%	1.52%	**1.15%**	1.00%	1.31%	1.56%	**1.29%**	**1.22%**
	1,000-1,500	2.55%	4.57%	4.10%	**3.74%**	3.08%	3.86%	4.90%	**3.95%**	**3.84%**
	1,500-2,000	6.76%	7.49%	7.31%	**7.18%**	6.85%	8.02%	9.12%	**8.00%**	**7.59%**
Mean		**3.36%**	**4.41%**	**4.31%**	**4.02%**	**3.64%**	**4.40%**	**5.19%**	**4.41%**	**4.22%**

Table 9
Item Stock Usage index of the model with random instances.

Item Stock Usage		Size of the Instances								Mean
		35 Item Types; 15 Object Types; 5 Final Products Types				70 Item Types; 30 Object Types; 10 Final Products Types				
		Item Inventory Costs								
		0-5%	10-15%	20-25%	**Mean**	0-5%	10-15%	20-25%	**Mean**	
Length of Items (mm)	500-1,000	14.6%	4.7%	4.4%	**7.9%**	14.5%	5.5%	4.0%	**8.0%**	**8.0%**
	1,000-1,500	19.5%	8.3%	5.5%	**11.1%**	20.1%	8.4%	5.3%	**11.2%**	**11.2%**
	1,500-2,000	11.8%	7.2%	8.4%	**9.1%**	15.9%	5.9%	6.1%	**9.3%**	**9.2%**
Mean		**15.3%**	**6.7%**	**6.1%**	**9.4%**	**16.8%**	**6.6%**	**5.1%**	**9.5%**	**9.4%**

Table 10
Processing time (in seconds) of the model with random instances.

Computational Time (s)		Size of the Instances								Mean
		35 Item Types; 15 Object Types; 5 Final Products Types				70 Item Types; 30 Object Types; 10 Final Products Types				
		Item Inventory Costs								
		0-5%	10-15%	20-25%	**Mean**	0-5%	10-15%	20-25%	**Mean**	
Length of Items (mm)	500-1,000	1345	1334	1334	**1338**	1732	1762	1780	**1758**	**1548**
	1,000-1,500	1296	932	960	**1063**	1505	1411	1453	**1456**	**1259**
	1,500-2,000	317	323	484	**375**	713	363	622	**566**	**470**
Mean		**986**	**863**	**926**	**925**	**1317**	**1179**	**1285**	**1260**	**1093**

interesting to mix items of different sizes allowing better combinations and reduced losses. This is a relevant managerial insight that can be used by the production planning sector.

Regarding the item stock usage index, shown in Table 9, as expected, instances with low inventory cost generated an increase in stock use. The size of instances and items does not show a clear trend in inventory levels. The values of the item stock cost index are, in general, slightly lower, with the total mean being 7.9%.

The same trend presented in Table 9 occur for the final products, both for stock usage index and stock cost index, for which the average values are 2.3% and 2.1%, respectively. For both items and final products, the cost index is lower than the usage index. This shows that the model was able, for random instances in general, to prioritize the formation of stocks of low-cost units.

The average use of the production capacity was 76.9%. In the first period, more capacity is used (84.9%), and the lowest use of capacity occurs in the last period (70.2%). In general, the use of the production capacity varies little with the variation of the parameters, therefore, only the total mean is presented, without the details for each group of instances. A slight trend that can be observed is that the higher the inventory costs, the less capacity was used. It is a natural result since, if inventory costs are high, inventory build-up is avoided, so fewer items are produced, and less production capacity is used. It is also observed that large instances tend to use a little more production capacity.

Table 10 shows the total computational time required, on average, for each instance group. As expected, this is directly related to the size of the instance. On the other hand, instances with smaller items generate greater possibilities for cutting patterns, so these instances require more iterations to optimize and, thus, require more computing time.

On average, 8.3 iterations were performed, and 335 columns were generated. Of course, the groups of instances that require more iterations are those that generate the largest number of columns and take the longest computational times.

All instances with small items have reached the 20 minutes limit to solve the integer master problem, on the other hand, for instances with large items this has occurred only a few times.

For these random instances, most of the processing time of the column generation was required to solve the subproblems (average time of 114.4 s). The average time needed to solve the relaxed master problem is 95.6 s. This difference is much greater analyzing only small instances. Therefore, regarding the column generation, it is possible to note that increasing the size of the instance has a greater impact on the processing time of the relaxed master problem. Accordingly, in Section 6.1, for the real data instance, 82.6% of the column generation time was demanded by the relaxed master problem.

According to the results, in general, the best results occur for smaller instances, and/or with large items. Instances with small items become more complex to be solved, so they generate worse results, except for the loss. Among the measures analyzed, the loss was the one that best showed the influence of changing each parameter. The utilization of production capacity, however, varies very little among different groups of instances.

7. Conclusion and future proposals

In this paper, an automotive spring factory was studied in order to reduce its inventory costs and losses in the steel bar cutting process. A mathematical model that captured practical issues of the company was developed to represent this problem. To analyze the performance of the model, tests with both real and random data were made. Results show that the proposed model worked well since, applied to real instances, it found significantly superior solutions to those achieved in practice. Also, the model was successfully validated by solving 180 instances with random data.

For its practical application, solutions were achieved in acceptable computational times, saving time and money for the company, while respecting all the specific operational restrictions and guaranteeing the applicability of the solution. In addition, losses were reduced in 49.7% using the proposed solution, generating great savings in raw materials, about 3.3 ton of steel per week. Besides the economic aspects, it is worth noticing the ecological impact of such reduction.

Solving random instances was important to better understand specific aspects of the problem, enabling managers to make better decisions. It also demonstrates the robustness of the model by solving 18 different types of instances, explaining, for several factors, the influence of parameter changes. The loss was the measure that best showed the influence of the variation of each parameter. In general, in terms of solution quality (gap), better results were found with smaller instances and/or with large items. It gives a managerial insight in the sense that, in practical cases, if one considers a subset of relatively larger items, the approach proposed in this paper will reach solutions of improved quality.

Analyzing the limitations of the approach proposed in this paper, it can be stated that the non-consideration of setups makes it difficult to apply the model in companies that use cutting machines with relevant setup costs, such as milling cutter, punching machines and other machines that require large tool changes from one object to another. In the same way, companies with high inventory costs would require adjustments for a satisfactory operation of this model. This may be the case for companies that have expensive processes downstream of the cutting process, or that produce items with high added value. In this case, the cost of loss of raw material is not as significant as the cost of a high stock of final products. Finally, modifications would be required to implement this model in environments where the cost of using productive capacity is more relevant, such as when using or providing outsourced services. In this case, the cost of material waste competes with the cost of using capacity.

As future research, a focus on medium/long term issues in the factory is suggested, by considering demand for objects, their stock limits and also the purchase of raw materials as a variable, dealing with the 3-level integrated lot sizing and cutting stock problem. Considering these new aspects, improvement on the solution approaches might be necessary, such as, stabilization techniques, reformulations and heuristic procedures to find an integer solution. It is also possible to analyze the performance of the approach used in other companies, whose production process is similar. Regarding the application in some other contexts, for a satisfactory result, it is necessary to include in the model the setup times of the cutting machines. In addition, the production capacity of the cutting process can be considered as a parameter, enabling an in-depth analysis of possible advantages of increasing or reducing capacity, since more capacity would lead to more idleness, but more freedom to cut on the preferred date.

Declaration of Competing Interest

None.

Acknowledgments

The autors would like to thank Molas Fama, Universidade Tecnológica Federal do Paraná (UTFPR), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP) for the support on this work. This research was funded by CNPq (Process numbers 421130/2018-0, 305261/2018-5, 306558/2018-1 and 406335/2018-4) and FAPESP (Process number 2013/07375-0 and 2016/01860-1).

References

[1] R.L. Torres, in: A indústria automobilística brasileira, Santa Catarina Federal University, Florianópolis, 2011, p. 179.

[2] National Confederation of Industry of Brazil. 2019. *A Indústria em Números: Julho de 2019*. Technical report. p. 8, v. 2.
[3] L.V. Kantorovich, Mathematical methods of organizing and planning production, Manage. Sci. 6 (1960) 366–422 vn. 4. https://doi.org/10.1287/mnsc.6.4.366.
[4] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting stock problem – part II, Oper. Res. 11 (1963) 863–888 v.n. 6. https://doi.org/10.1287/opre.11.6.863.
[5] G. Wäscher, H. Haußner, H. Schumann, An improved typology of cutting and packing problems, Eu J. Oper. Res. 183 (2007) 1109–1130 v.n. 3. https://doi.org/10.1016/j.ejor.2005.12.047.
[6] M. Hifi, Special issue: cutting and packing problems, Stud. Inform. Universalis 2 (2002) 1–161 v.n. 1.
[7] Y. Pochet, L. Wolsey, A Production Planning by Mixed Integer Programing, Springer, New York, 2006.
[8] D.J. Fiorotto, R. Jans, S.A. de Araujo, An analysis of formulations for the capacitated lot sizing problem with setup crossover, Comput. Ind. Eng. 106 (2017) 338–350 v. https://doi.org/10.1016/j.cie.2016.12.037.
[9] J.T. Wong, C.T. Su, C.H. Wang, Stochastic dynamic lot-sizing problem using bi-level programming base on artificial intelligence techniques, Appl. Math. Modell. 36 (2012) 2003–2016 v.n. 5. https://doi.org/10.1016/j.apm.2011.08.017.
[10] W.H. Ho, Y.H. Chiu, Y.J. Chen, Multi-objective Pareto adaptive algorithm for capacitated lot-sizing problems in glass lens production, Appl. Math. Modell. 53 (2018) 731–738 v. https://doi.org/10.1016/j.apm.2017.09.030.
[11] H. Adeinat, J.A. Ventura, Integrated pricing and lot-sizing decisions in a serial supply chain, Appl. Math. Modell. 54 (2018) 429–445 v. https://doi.org/10.1016/j.apm.2017.09.044.
[12] M. Mohammadi, M. Esmaelian, A. Atighehchian, Design of mathematical models for the integration of purchase and production lot-sizing and scheduling problems under demand uncertainty, Appl. Math. Model. (2020). https://doi.org/10.1016/j.apm.2020.03.021.
[13] K.C. Poldi, in: O problema de corte de estoque multiperíodo, University of São Paulo. Institute of Mathematical and Computer Sciences, 2007, p. 109.
[14] M.C.N. Gramani, P.M. França, M.N. Arenales, A Lagrangian relaxation approach to a coupled lot-sizing and cutting stock problem, Int. J. Prod. Econ. 119 (2009) 219–227 v.n. 2. https://doi.org/10.1016/j.ijpe.2009.02.011.
[15] M. Vanzela, G.M. Melega, S. Rangel, S.A. de Araujo, The integrated lot sizing and cutting stock problem with saw cycle constraints applied to furniture production, Comput. Oper. Res. 79 (2017) 148–160 v. https://doi.org/10.1016/j.cor.2016.10.015.
[16] G.M. Melega, S.A. de Araujo, R. Jans, Classification and literature review of integrated lot-sizing and cutting stock problems, Eu. J. Oper. Res. 271 (2018) 1–19 v. https://doi.org/10.1016/j.ejor.2018.01.002.
[17] M.C.N. Gramani, P.M. França, M.N. Arenales, A linear optimization approach to the combined production planning model, J. Franklin Inst. B 348 (7) (2011) 1523–1536 v. https://doi.org/10.1016/j.jfranklin.2010.05.010.
[18] K.C. Poldi, M.N. Arenales, O problema de corte de estoque unidimensional multiperíodo. Pesquisa Operacional 30 (1) (2010) 153–174 v. https://doi.org/10.1590/S0101-74382010000100008.
[19] T. Wu, K. Akartunali, R. Jans, Z. Liang, Progressive selection method for the coupled lot-sizing and cutting-stock problem, INFORMS J. Comput. 29 (3) (2017) 523–543 v. https://doi.org/10.1287/ijoc.2017.0746.
[20] S.C. Poltroniere, K.C. Poldi, F.M.B. Toledo, M.N. Arenales, A coupling cutting stock-lot sizing problem in the paper industry, Ann. Oper. Res. 157 (1) (2008) 91–104 v. https://doi.org/10.1007/s10479-007-0200-6.
[21] K.C. Poldi, S.A. de Araujo, Mathematical models and a heuristic method for the multiperiod one-dimensional cutting stock problem, Ann. Oper. Res. 238 (1-2) (2016) 497–520 v. https://doi.org/10.1007/s10479-015-2103-2.
[22] A.A.S. Leão, M.M. Furlan, F.M.B. Toledo, Decomposition methods for the lot-sizing and cutting-stock problems in paper industries, Appl. Math. Modell. 48 (2017) 250–268 v. https://doi.org/10.1016/j.apm.2017.04.010.
[23] S.G. Santos, S.A. de Araujo, M.S.N. Rangel, Integrated cutting machine programming and lot sizing in furniture industry, Pesquisa Operacional para o desenvolvimento 3 (1) (2011) 1–17 v.
[24] G.M. Melega, S.A. de Araujo, R. Morabito, Mathematical model and solution approaches for integrated lot-sizing, scheduling and cutting stock problems, Ann. Oper. Res. (2020) 1–42. https://doi.org/10.1007/s10479-020-03764-9.
[25] C.T.L.S. Ghidini, D. Alem, M.N. Arenales, Solving a combined cutting stock and lot-sizing problem in small furniture industries, in: Proceedings of the 6th International Conference on Operational Research for Development (VI-ICORD), 2007.
[26] S.M.A. Suliman, An algorithm for solving lot sizing and cutting stock problem within aluminum fabrication industry, in: Proceedings of the 2012 International Conference on Industrial Engineering and Operations Management, 2012, pp. 783–793.
[27] D.J. Alem, R. Morabito, Production planning in furniture settings via robust optimization, Comput. Oper. Res. 39 (2012) 139–150 v.n. 2. https://doi.org/10.1016/j.cor.2011.02.022.
[28] D.J. Alem, R. Morabito, Risk-averse two-stage stochastic programs in furniture plants, OR Spectrum 35 (2013) 773–806 v.n. 4. https://doi.org/10.1007/s00291-012-0312-5.
[29] A.A. Farley, Mathematical programming models for cutting-stock problems in the clothing industry, J. Oper. Res. Soc. 39 (1988) 41–53 v.n. 1. https://doi.org/10.1057/jors.1988.6.
[30] F.K. Lemos, A.C. Cherri, S.A. de Araujo, The cutting stock problem with multiple manufacturing modes applied to a construction industry, Int. J. Prod. Res. (2020) 1–19. https://doi.org/10.1080/00207543.2020.1720923.